

A ReRAM Memory Compiler for Monolithic 3D Integrated Circuits in a Carbon Nanotube Process

EDWARD LEE, DAEHYUN KIM, JINWOO KIM, SUNG KYU LIM, and SAIBAL MUKHOPADHYAY, Georgia Institute of Technology, USA

We present a ReRAM memory compiler for monolithic 3D (M3D) integrated circuits (IC). We develop ReRAM architectures for M3D ICs using 1T-1R bit cells and single and multiple tiers of transistors for access and peripheral circuits. The compiler includes an automated flow for generation of subarrays of different dimensions and larger arrays of a target capacity by integrating multiple subarrays. The compiler is demonstrated using an M3D process design kit (PDK) based on a Carbon Nanotube Transistor technology. The PDK includes multiple layers of transistors and back-end-of-the-line integrated ReRAM. Simulations show the compiled ReRAM macros with multiple tiers of transistors reduces footprint and improves performance over the macros with single-tier transistors. The compiler creates layout views that are exported into library exchange format or graphic data system for full-array assembly and schematic/symbol views to extract per-bit read/write energy and read latency. Comparison of the proposed M3D subarray architectures with baseline 2D subarrays, generated with a custom-designed set of bit cells and peripherals, demonstrate up to 48% area reduction and 13% latency improvement.

CCS Concepts: • **Hardware** → **Memory and dense storage; Memory compiler;**

Additional Key Words and Phrases: Memory compiler, monolithic 3D integration, ReRAM

ACM Reference format:

Edward Lee, Daehyun Kim, Jinwoo Kim, Sung Kyu Lim, and Saibal Mukhopadhyay. 2021. A ReRAM Memory Compiler for Monolithic 3D Integrated Circuits in a Carbon Nanotube Process. *J. Emerg. Technol. Comput. Syst.* 18, 1, Article 20 (November 2021), 20 pages.
<https://doi.org/10.1145/3466681>

1 INTRODUCTION

Monolithic 3D (M3D) integrated circuits (IC) allow fine-grain vertical integration of transistors allowing improved scalability, reduced wire-length, and ultimately, leading to better energy-efficiency. Compared to large **through-silicon-vias (TSVs)** used in conventional 3D-ICs [10, 14], the M3D-ICs use high-density inter-layer vias (ILVs) providing much higher routing resources [26, 30]. M3D enables the possibility of designing multi-layer macros where cells of macros can be partitioned in multiple tiers and connected via fine grid ILVs.

This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) (Grant No. HR001118C0096). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA.

Authors' address: E. Lee, D. Kim, J. Kim, S. K. Lim, and S. Mukhopadhyay, Georgia Institute of Technology, Atlanta, Georgia, USA; emails: elee359@gatech.edu, daehyun.kim@gatech.edu, jinwookim@gatech.edu, limsk@ece.gatech.edu, saibal.mukhopadhyay@ece.gatech.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

1550-4832/2021/11-ART20 \$15.00

<https://doi.org/10.1145/3466681>

In recent years, significant research efforts have been directed to develop **electronic design automation (EDA)** tools for partitioning, placement, and routing of logic in M3D technologies [15, 18, 20]. In comparison, there are relatively few works on memory design for M3D. Vertical BL SRAM arrays have been proposed to reduce BL capacitance to achieve improvement in both delay and power in multi-layer processes [19]. The impact of M3D design on in-memory computation has also been explored [29]. Although, these prior works show the feasibility of using M3D for memory design, to the best of our knowledge, there have been no prior on EDA tools for memory design M3D-ICs.

Resistive Random Access Memory (ReRAM) has emerged as an attractive solution for embedded non-volatile-memory for various applications including **vector matrix multiplication (VMM)** matrices [7, 22], **single-level cell (SLC)** memory [5, 6], **multi-level cell (MLC)** memory [4], and nonvolatile processors [16, 17] to name a few. The ReRAM devices can be formed with metal oxides [31] that allow the fabrication to be compatible with **back-end-of-line (BEOL)** processes. The BEOL integration brings opportunity of seamlessly incorporating ReRAM within 3D-compatible transistor technologies [8] to form multi-tier memory structures integrated with logic in a monolithic fashion [25, 27, 32]. The M3D integration of ReRAM and FETs have shown potential for many applications [27, 32]. More recently, commercial-scale fabrication technology and associated **process design kit (PDK)** have been demonstrated considering multiple layers of transistors and ReRAM devices [27].

Carbon Nanotube Transistor (CNT) technologies have emerged as a promising candidate for true M3D integration [26]. Temperature restrictions for sequential M3D fabrication often result in reliability and performance degradation in silicon-based M3D processes [2, 11, 24] and is a key challenge that has been explored in recent years [13, 21]. Meanwhile, the low temperature (<200°C) [24] process of transferring pre-fabricated CNTs allows a promising alternative to seamless M3D integration without introducing defects in lower-layer devices [3, 12, 23, 27]. When integrated with BEOL-compatible (<425°C) ReRAM [1, 24, 32] processes, this creates the many opportunities for designing new system architectures. However, designing such systems requires memory compilers for M3D ReRAM. This article, for the first time, to the best of our knowledge, presents a ReRAM memory compiler for M3D-ICs. The presented compiler uses custom designed 1T-1R cells and read/write peripheral circuits but automatically generates (scalable) ReRAM subarrays of different dimensions and micro-architectures. We present multiple ReRAM subarray architectures for M3D ICs including single and multiple tiers of transistor. **An advantage of Carbon Nanotube Transistor technologies is little cross-layer performance degradation across different transistor layers [27]. The multi-tier transistor subarray design builds upon this characteristic and entire peripheral blocks are “ported” and implemented with different-tier transistors from a single baseline design.** This allows exploration of different subarray structures to focus on M3D structural advantages by minimizing physical implementation change in individual blocks. In particular, we show the feasibility of fine-grain integration of access transistors and peripheral circuits in multiple tiers to reduce memory footprint and improve performance. Finally, we present an automated flow for integration of subarrays to generate physical design of a large-scale ReRAM in M3D-ICs.

The memory compiler is demonstrated considering layer structure of M3D-IC PDK presented by Srimani et al. [27] that includes two layers of ReRAM, two layers of transistors, and eleven metal layers (Figure 1). The back-gate transistor structure reduces gate-to-plug capacitance [28], which can greatly benefit memory design in terms of both access latency and energy. We consider 1T-1R ReRAM bit cells, and associated custom designed peripheral circuits. We demonstrate compilation of multiple types of ReRAM arrays, namely, (1) **STF-ReRAM**: Single tier of FETs containing access transistors and peripheral circuits, (2) **MTF-BL-ReRAM**: Multiple tiers of FETs

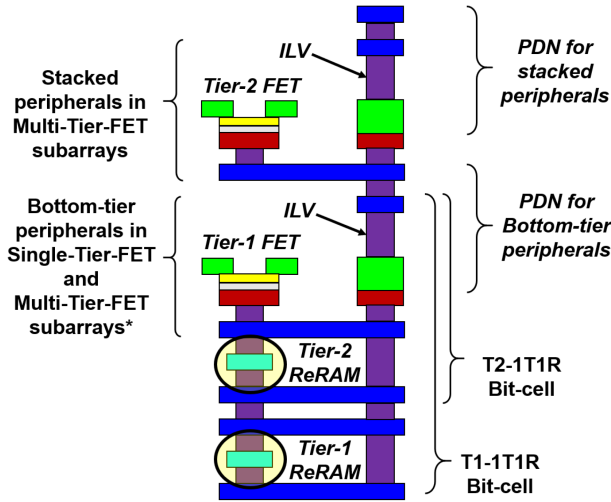


Fig. 1. Layer organization of M3D PDK and usage in the subarray generation for 1T1R bit cells and peripherals (re-drawn after Reference [27]).

with Bit-line peripherals in a second layer of transistors, and (3) **MTF-ALL-ReRAM**: Multiple tiers of FETs with **all** peripherals in one layer of transistors and access transistors in a different layer. The compiler creates layout views (**library exchange format (LEF)** and **graphic data system (GDS)**) and schematic/symbol views of memory modules. The memory designed with MTF subarrays show lower footprint, higher cell density, and reduced read latency/energy compared to the ones designed with STF subarray.

This article makes the following key contributions:

- Design and implement a fully automated ReRAM compiler in a CNT M3D-IC PDK with layout-precise performance evaluation and scalable array generation for **Design Space Exploration (DSE)**.
- Demonstrate and explain the key benefits of M3D subarrays with fine grid ILVs.
- Implement and analyze three different subarray structures: STF-ReRAM, for possible logic-on-memory applications(baseline), and two high-density multi-tier transistor subarrays MTF-BL-ReRAM and MTF-ALL-ReRAM.
- Implement and compare a baseline 2D subarray structure in a silicon technology.
- Demonstrate and quantify the automated tool-flow and techniques for increased evaluation speed.
- Explore design space of different subarray dimensions and structures to evaluate and exploit the benefits of the multi-tier transistor + ReRAM PDK.

2 MONOLITHIC 3D RERAM DESIGN

Multi-tier designs of M3D subarrays that are implemented and analyzed in this work focus on an approach similar to citation [29] where the bit cells remain in a 2D arrangement. But instead of utilizing a second layer for assisting cell-level operations, entire peripherals are folded and implemented in T2 FETs to minimize footprint (Figure 2). This design topology is made possible by the unique Monolithic 3D integrate-ability of carbon nanotubes transistors, which alleviates performance loss in upper-layer transistors in silicon-based M3D technologies. Different peripherals blocks are placed in T2 FETs depending on Multi-tier structures proposed and discussed in later

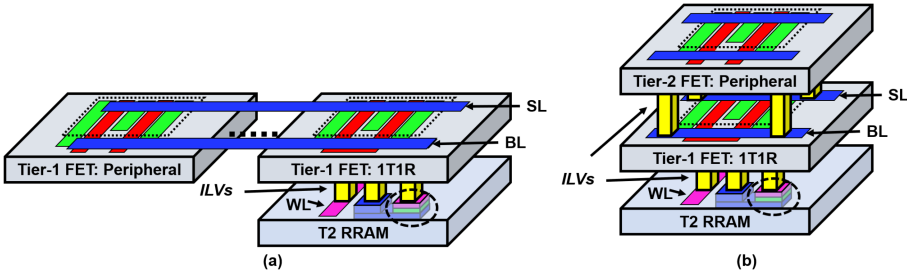


Fig. 2. Simplified visualization of folding/stacking peripheral circuit blocks to upper Tier-2 (T2) FETs to reduce footprint. (a) Baseline STF subarray. (b) Stacked peripherals MTF subarray.

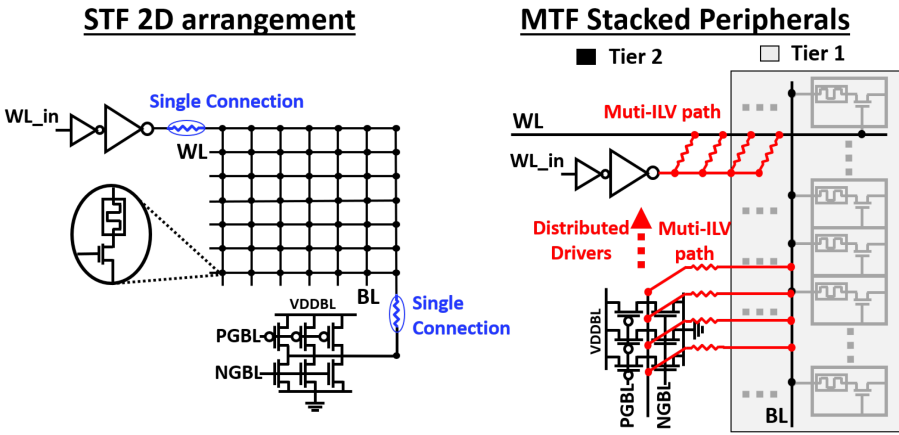


Fig. 3. Stacking peripherals with multiple ILVs compared to single-tier 2D architectures.

sections. This also allows fine-grid ILV connections to reduce I-R drop during high-current programming of ReRAM devices. Following discussions will use **Single-tier-FETs (STF)** to refer to designs where all transistors used are in the same tier, and **Multi-tier-FETs (MTF)** for designs where transistors in multiple tiers are used.

2.1 Multi-tier Peripheral Circuits

The M3D design allows transistors in the peripheral circuits to be in the same layer as the access transistor similar to a traditional 2D design (STF-ReRAM). However, a major advantage of the M3D technology is to be able to distribute peripheral circuits across multiple tiers (MTF-ReRAM). The word-line and/or bit-line drivers are placed in a different tier from the access transistor and multiple ILVs are used to connect the peripherals to the active lines (WL/BL/SL) in the ReRAM crossbar. The most obvious benefit of MTF-ReRAM is a reduction in the footprint. The memory crossbar only has single transistor per cell (1T-1R). However, as ReRAMs require large write current, the single device must have large driving capacity (size), which in turn requires large devices in peripheral circuits. Hence, peripheral circuits can occupy appreciable physical area, which can be reduced via MTF-ReRAM design. This is particularly beneficial for smaller subarray dimensions as illustrated later in Section 5.

The MTF-ReRAM also provide additional performance (and robustness) benefits. The multiple ILVs connecting the BL/WL drivers and interconnects within the crossbar allows parallel paths

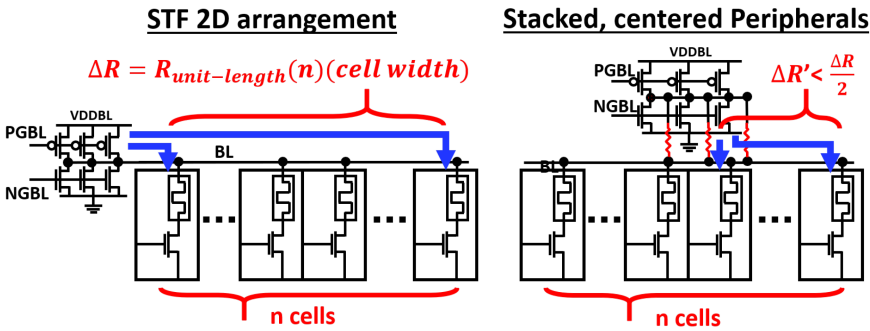


Fig. 4. Centering stacked peripherals to reduce worst-case mismatch seen by bit cells along the same BL.

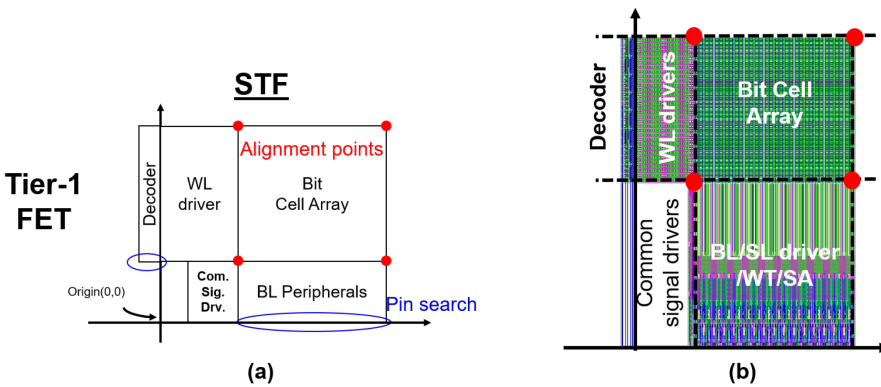


Fig. 5. (a) Layout Arrangement of baseline STF subarray. (b) Top view of STF subarray.

that can reduce IR drop through the WL/BL (Figure 3). This is especially critical for programming ReRAM devices as the instantaneous current can be high when the device is in low-resistance-state(LRS). This benefit can be extended even further by distributing BL drivers, which allow even more ILVs to be placed in parallel. As an example, **post-parasitic-extraction (post-PEX)** simulations of $64\text{WL} \times 64\text{BL}$ subarrays showed stacking peripherals can reduce IR drop during write by 21 mV and by further distributing drivers, IR drop reduction becomes 29 mV. Further, the placement of peripherals across multiple tiers can be used to reduce mismatch seen by bit cells in the array (Figure 4). The post-PEX simulation of the $64\text{WL} \times 64\text{BL}$ subarray shows that placing the peripherals to center output of drivers on top of the bit-cell array, worst-case mismatch in IR drop across the BL can be reduced from ≈ 40 mV to ≈ 20 mV. It is imperative that reduced magnitude (and mismatch) of IR drop also improves the voltage margin (and its uniformity) at the bit cells, thereby improving the overall robustness. However, creating distributed drivers can result in overhead in access energy due to the extended length and extra parasitics as discussed in Section 5.

2.2 M3D Subarray Micro-architectures

We design three types of micro-architectures for scalable and automated design of subarrays. The first option is a 2D implementation of bit cells surrounded by peripheral circuits on each side. This structure is named as **Single-tier-FETs (STF)** subarrays. The arrangement of this architecture is shown in Figure 5. The 1T1R bit cells were arranged in such that SL is parallel with BL, and both perpendicular to WL. The peripherals are implemented as single blocks surrounding the cell array.

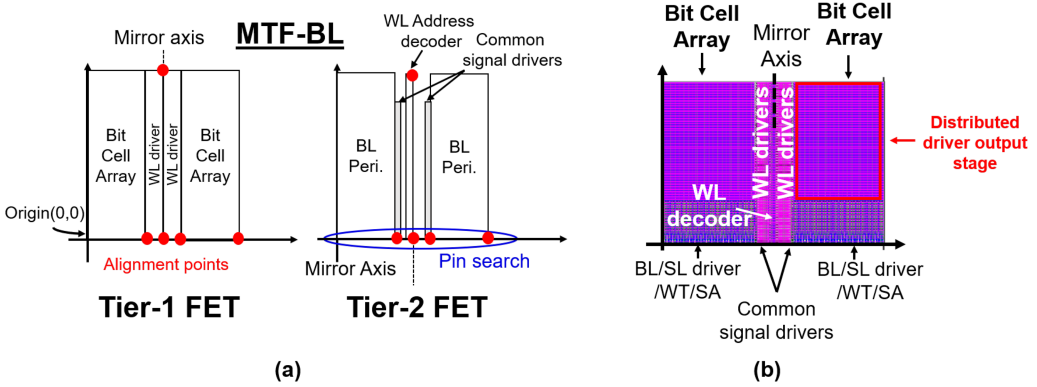


Fig. 6. (a) Layout Arrangement of baseline MTF-BL subarray. (b) Top view of MTF-BL subarray.

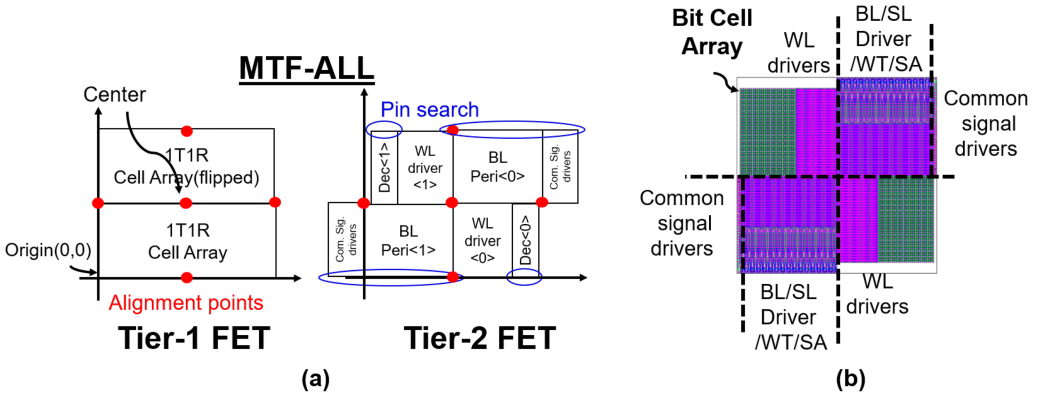


Fig. 7. (a) Layout Arrangement of baseline MTF-ALL subarray. (b) Top view of MTF-ALL subarray.

The second type of subarray architecture is named as **Multi-tier-FETs-BL (MTF-BL)**, where the stacking of BL peripherals is emphasized. Figure 6 shows the structure of this design. The WL drivers are placed in Tier-1 along with bit cells, whereas WL address decoders and BL peripherals are in Tier-2. This structure implements distributed drivers with additional ILV connections to BL to minimize IR drop during write. The strength of the driver output stage has a fixed upper-limit, but can be reduced when BL lengths are limited by small subarray dimension. The mirrored structure allows all access pins to be to the same edge while stacking peripherals.

The third type of subarray architecture is named as MTF-ALL (Figure 7). In this design, the BL peripherals and decoders as well as the WL peripherals are placed in the second tier of transistors. The arrangement is such that, with peripherals and bit cells rotated/flipped, it can ensure that output edges of peripherals are aligned along the center axes of the bottom bit-cell arrays. This structure prevents BL peripherals from blocking WL access and vice versa by segmenting all peripheral instantiating to twice. Centering the output of BL/WL peripherals also reduces mismatch in interconnect lengths, and hence, IR drops. However, the central location of peripherals also separates the access pins to different edges, which needs to be considered during automated generation for multiple inter-connected subarrays.

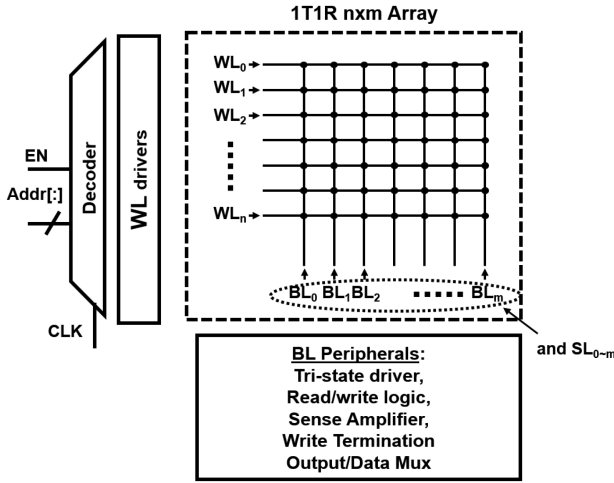


Fig. 8. Building blocks of the subarray.

3 SUBARRAY BUILDING BLOCKS

To demonstrate the Monolithic-3D design capability of the ReRAM memory compiler, a set of custom building blocks are designed and implemented. The digital friendly circuit topologies (and custom layouts) are used to support automated generation of physical design of subarrays of varying dimensions.

3.1 1T-1R Bit Cells

We consider one-transistor one-resistor (1T-1R) ReRAM bit cells for this work as 1T-1R bit cells are the most commonly used ReRAM bit cell in literature for high control-ability. While the access transistor introduces area overhead, it eliminates the need for half-select lines [9] required to prevent write disturbance in neighboring cells, which guarantees best access robustness.

Layer usage of supported 1T-1R bit cells are also shown in Figure 1. T2-1T1R cells use less metal stack than T1-1T1R cells. While the both types of bit cells are supported in the compiler, for sake of simplicity, this article will focus on T2-1T1R, where top-tier (T2) ReRAMs are used, for array generation, simulation, and performance analysis.

3.2 ReRAM Read/Write Circuitry

ReRAM devices have certain characteristics (such as device resistance/write-time variations) that demand additional devoted control circuitry [6, 17] to create robust access of bits when used as memory bit cells as opposed to SRAMs and DRAMs. The pre-designed bit-line peripherals within the subarray generator used for demonstration are therefore also designed to include this functionality. Figure 8 shows the building blocks of the subarrays created. Figure 10 shows a simplified schematic of the periphery circuits that directly access bit cells implemented in our ReRAM compiler.

Word-line (WL) peripherals. The WL driver is custom designed as a 4-stage buffer pitch-matched to the 1T1R cell-widths. The WL address decoder is created based on three basic cells, a 1-to-2 decoder, a 2-to-4 decoder, and a 2-to-4 decoder with **clock input (CLK)**, which is used for the final stage. The design strategy assumes addresses for bit-cell access arrive before the CLK is pulled high to enable WL drivers for access. Figure 9(a) shows a schematic of the basic cells (the 2-to-4

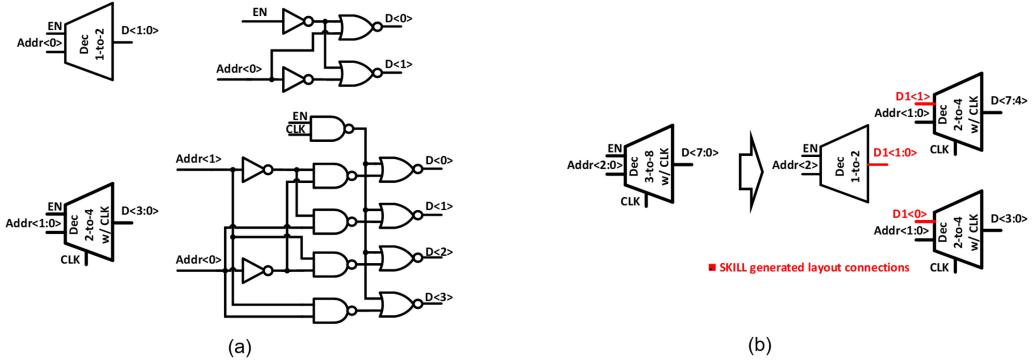


Fig. 9. (a) Basic decoder blocks and schematics. (b) 3-to-8 decoder example of arbitrary decoder size generation.

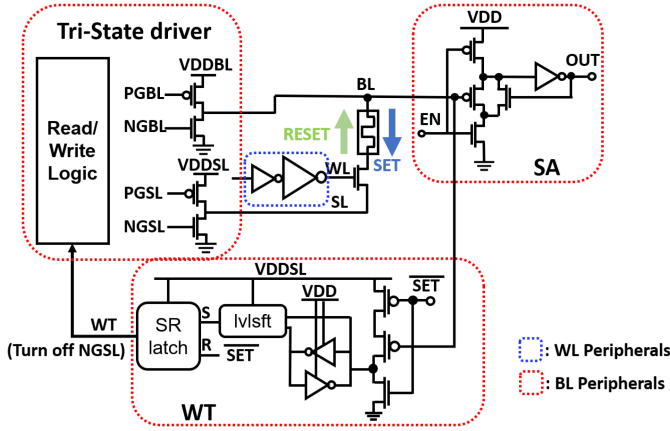


Fig. 10. Schematic of bit-cell access.

decoder simply replaces the CLK input NAND with an inverter for EN only and is not shown). Any arbitrary bus width that is a power of 2 can be derived from these cells in an iterative manner. This is automated in SKILL to ensure scalability of subarray generation within the memory compiler. Figure 9(b) shows the formation of higher bit width decoders with lower bit width decoders with an example of a 3-to-8 decoder. The decoder basic cell layouts are custom designed to ensure automated wire connections with SKILL are DRC/LVS clean.

Bit-line (BL) Peripherals—Write Circuits. The write peripheral circuits apply the large voltage and support bi-directional current through the bit cells [17]. As discussed in Section 2.1, a critical advantage of M3D subarrays is the ability to address IR drop across bit lines when large programming current is consumed. Design of write circuits will also need to account for similar considerations. To mitigate IR-drop overhead from either transmission gates or stacked transistors used in typical tri-state buffers (used to decouple write circuits during read operations), a tri-state driver design with separate pull-up/pull-down controls is implemented. The PGBL/NGSL is ON for SET (HRS→LRS) and PGSL/NGSL is ON during RESET (LRS→HRS) to apply opposite polarity voltage across the bit cells for each case. During IDLE state, the BL/SL is pulled to zero through ON NGBL/NGSL. During read, PGBL/PGSL are both pulled high to allow BL to be pre-charged

Table 1. Simplified Truth Table for Read/Write Logic

State	PGBL	NGBL	PGSL	NGSL
IDLE	1	1	1	1
READ	1	0	1	1
SET	0	0	1	1 \rightarrow 0
RESET	1	1	0	0

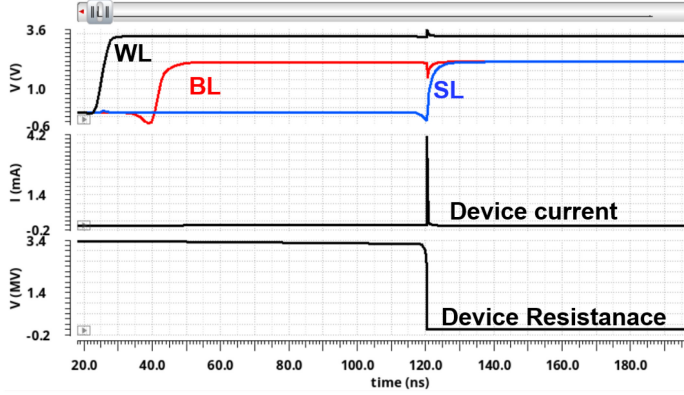


Fig. 11. Simulation waveform of “SET” termination.

at lower voltage (prevent read disturbance), and $NGSL = 1$ allows discharge through SL during sensing. A simplified truth table of the read/write logic shown in Figure 10 is shown in Table 1. Write termination refers to the operation of sensing and cutting off write operation when the device has been programmed to a specific threshold; this reduces extensive energy waste due to high write current [17] at LRS states. Analog write termination circuits often involve the addition of an extra pull-down path for writing, which can create large area overhead when device strengths are weak [6]. In the compiler, the baseline design implemented for demonstration uses a digital driver-based write termination circuit (similar to Reference [17]). When the device is successfully “SET,” the **bit-line (BL)** voltage reduces slightly due to the decrease in ReRAM resistance, which triggers the back-to-back inverter to switch. This signal is level-shifted to VDD_{SL} , latched and is used to disable writing by switching off the pull-down path through SL (switching $NGSL$ to 0). Figure 11 shows a sample simulation waveform of this operation. It can be observed that device resistance gradually reduces after BL is pulled high. After the device resistance drops (successfully “SET”ted), the droop in BL voltage is sensed by the WT circuit and SL voltage increases after $NGSL$ is turned off. The increased device current when the bit cell switches from HRS to LRS is thus cut off and the write operation is terminated.

Bit-line Peripherals—Read Circuits. The bit-line peripherals also include a sense amplifier to read the data stored in the bit cells. PGBL is pulled high and NGBL pulled low to leave BL floating for precharge and discharge during sensing. Once the WL signal is raised high, the BL voltage drops through ReRAM resistance and an access transistor. The rate of discharge of BL can be used to determine the state of the ReRAM resistance (LRS or HRS). Once the BL drops below a threshold, p-FET discharging transistor turns on, and a positive feedback is implemented to increase the switching speed. The pulse width of the enable signal is controlled to ensure only fast

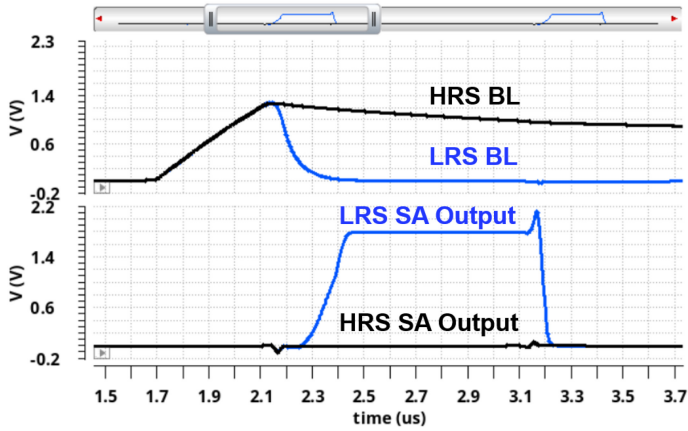


Fig. 12. Simulation waveform of Read for HRS and LRS bit cells.

discharge of the BL, due to low-resistance of the ReRAM, allows switching of the sense-amplifier. Figure 12 shows a sample simulation waveform of reading bit cells programmed to HRS and LRS states, respectively.

4 M3D RERAM COMPILER

The M3D ReRAM compiler consists of two components. First, generation of ReRAM subarrays for given dimensions (i.e., number of rows and number of columns), and second, integration of multiple subarrays to create a memory array of a target capacity. The performance of various subarrays and the memory array are evaluated as a part of the compiler flow.

4.1 Subarray Compilation and Evaluation

Figure 13 shows a flow chart describing the subarray generator. The subarrays are created by integrating custom design blocks while following certain subarray structures. The different peripheral circuits are maintained and can be used during subarray compilation without affecting the overall generation flow. The subarray generation relies heavily on script-based layout generation, such as SKILL, to support this functionality and generate physical design of different dimensions. Along with generating the physical design and netlist, the flow also generates schematics and symbols for circuit assembly, diagnosis and evaluation. Schematic and symbols are saved in a design library and layouts are exported as abstract LEF files and GDS files.

Figure 13(b) shows the flow for energy and latency estimation. To provide post-layout performance evaluation, **parasitic extraction (PEX)** is integrated into the evaluation flow. However, extracting parasitics for the full subarray would demand high amount of resources while producing unnecessary overhead. Instead, we extract the only the critical bit cells and its neighboring rows and columns to account for idle cells and BL/WL coupling capacitance. For this purpose, a simplified layout from the full subarray is created, which only contains four rows and four columns. This structure greatly reduces the netlist complexity while ensuring that parasitics from adjacent lines are included along the worst-case path. To reduce simulation time by a step further, the tool internally replaces idle (not accessed in testbench) cells with dummies containing only transistor and no ReRAM (1T0R cells). Only the far-end corner contains 1T1R bit cells for extraction. This greatly reduces convergence time consumed in following transient simulations, where solving exponential expressions used to model ReRAM device behavior in unselected cells are a

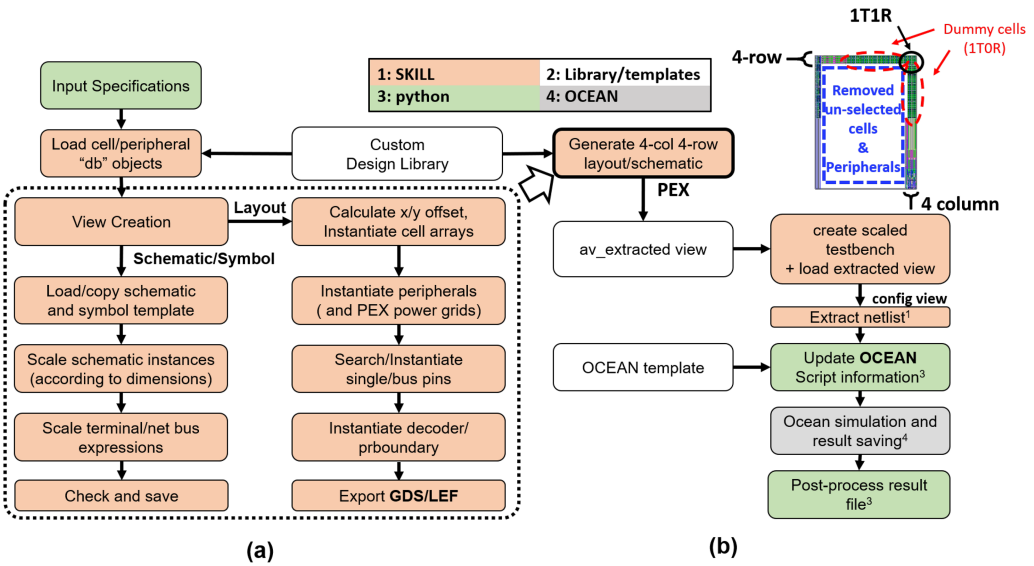


Fig. 13. Subarray compiler: (a) layout/schematic/symbol generation; (b) performance evaluation and sample four-column/four-row subarray.

large overhead. The combined impact of these procedures reduces overall evaluation time by $>8\times$ from an impractical simulation period to acceptable ranges (detailed compilation time explained in Section 5). PEX is done with Quantus QRC Extraction with rules provided within the PDK, which generates an “av_extracted” cell view as indicated in Figure 13.

A testbench (“config view” in Figure 13(b)) is used for layout-precise estimation of latency and energy. The pre-set transient simulation template involves a write followed by a read. The power drawn from sources during the simulation is integrated to estimate energy. The contributions from WL and BL peripherals are decoupled to assist in design optimization. Read latency is measured as the delay from the point when decoder output is enabled to the final switching of the SA. Write latency is determined by the time of WL arrival to the time internal resistances are switched. The energy and latency data are saved for array-level design decisions where subarrays are instantiated as hard macros with flopped input/output ports for single-clock access. This approach of allowing flip-flops to mask subarray access prevents the need to create liberty files for the analog read/write operation, while guaranteeing access robustness when sufficient margin and constraints are applied during the array integration step. Note, as the write latency is almost entirely determined by the ReRAM device characteristics, it is largely unaffected by the subarray architecture and compiler flow. Hence, we do not discuss write latency in the following compilation/simulation results.

4.2 Memory Array Generation

Full memory array assembly is implemented using subarray macros along with generated array-level control logic in a standard 2D **place and route (PNR)** flow. To allow for design space exploration and to enable the ability to optimize for different time/area/power targets, a scalable H-tree architecture is adopted and implemented (Figure 14). The leaf node is the subarrays with peripherals. The higher nodes consist of the maximum four lower nodes and are integrated with decoders and **multiplexers (MUXs)**. Each hierarchy is divided by **flip-flops (F/Fs)** and constructs the pipeline architecture to meet a target frequency.

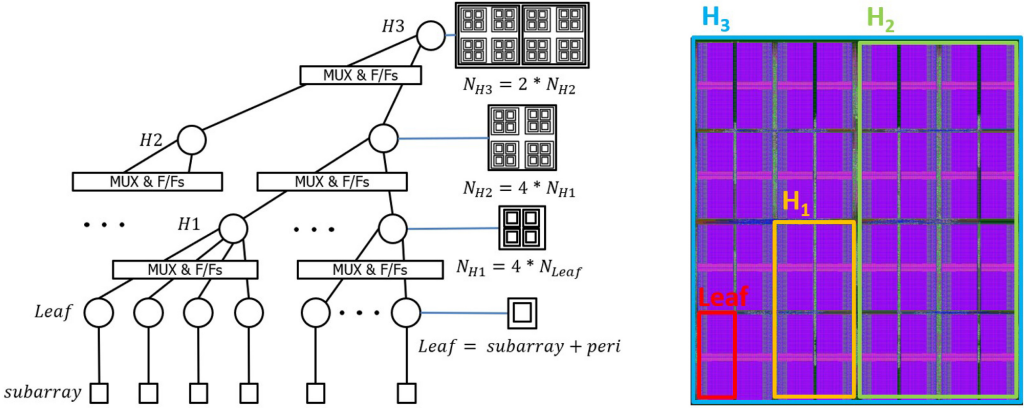


Fig. 14. Memory array generation: (a) H-tree architecture and (b) components in array layout.

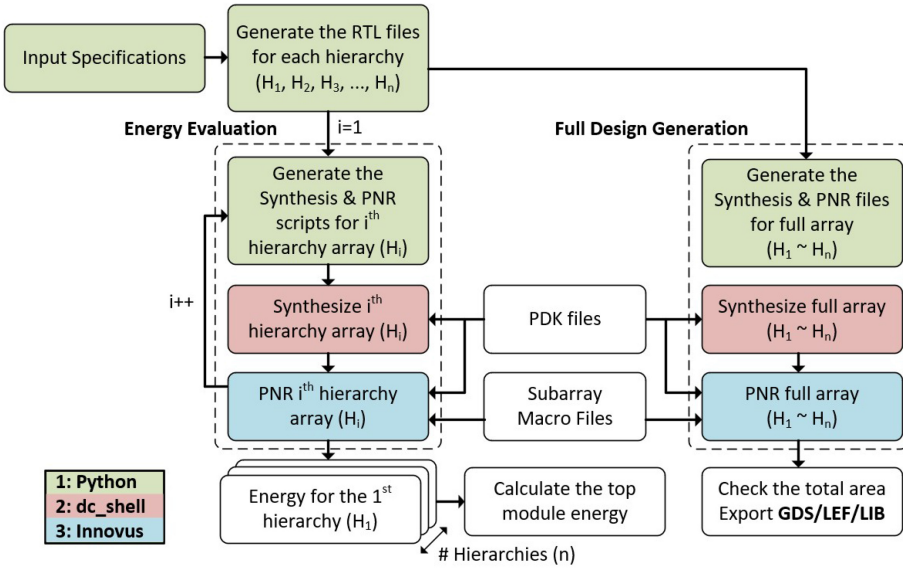


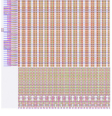

Fig. 15. Basic flow of generating full array and evaluate the top module properties.

The maximum frequency of the array is determined mainly by the access latency of subarrays (estimated from the subarray compiler). The number of levels in the H-tree must ensure that the maximum frequency of the H-tree access path is more than the maximum frequency of the subarrays. Once determined the array latency is defined by the number of levels in the H-tree.

Given a memory address, only a single subarray is active at a time, and only one node of the H-tree is active at each hierarchy. All other subarrays and nodes of the H-tree are in the idle state. Hence, to estimate array energy, we run the EDA tools for each hierarchy’s RTL file separately. For example, in Figure 14(b), we extract the energy from each hierarchy H_1 , H_2 , and H_3 . After we extract the energy from each hierarchy, we add them all to calculate the total energy consumption in the top module considering the subarray activity.

Figure 15 shows the flow chart for the array generation. The hierarchical H-tree connections between subarrays are done by EDA tools for 2D place and route. The loaded subarray macros are

Table 2. Benchmark Comparison of Technology PDK Along with Simulated Performance and Layout of Custom-designed Subarrays

Technology PDK	CMOS 130nm	CNT 130nm [27]	Subarray Type	CMOS 130nm, 2D	CNT 130nm, 2D(STF)
FET ON-OFF ratio	$> 10^6$	$\sim 10^4$	32WL×32BL Subarray Layout (to-scale)		
ReRAM ON-OFF ratio	-	$\sim 10^4$			
ReRAM SET voltage	-	~ 0.8 V			
ReRAM RESET voltage	-	~ -1 V			
32WL×32BL 2D Subarray (Normalized to CNT design)					
Read Latency	4.37*	1			
Read Current ratio	28.8	1	Footprint	0.499	1
Read Energy	0.79	1			
Write Energy	0.63	1			
* Not Optimized					

treated as block boxes that have strict input/output timing constraints to ensure close-proximity instantiation of masking flip-flops. The full-array design starts with the generation of **Register-transfer level (RTL)** files for each hierarchy. Next, RTL files of all hierarchies are used simultaneously, to generate the GDS, LEF, and LIB files for the full-chip design. Finally, we evaluate the energy of the entire array by aggregating array control power needed for subarray leaf access and the subarray access energy.

5 COMPILATION RESULTS

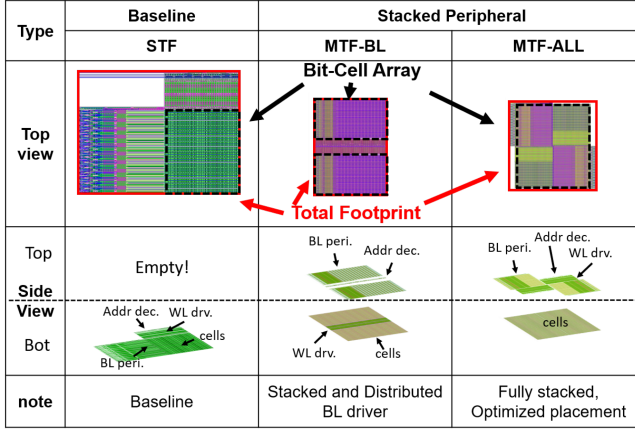
5.1 Technology Bench-marking

To characterize the subarrays designed in the CNT PDK, we have implemented a similar 2D design in a commercial 130 nm CMOS PDK for comparison. Table 2 shows detailed metrics investigated in this analysis. Same ReRAM device models from the CNT PDK are used in simulations to ensure an un-biased evaluation. To insert the device models for post-layout performance evaluation even when the CMOS PDK does not provide ReRAM devices, additional post-processing is integrated. Only 1-T devices are layed-out in the subarray. However, the subarray schematic and layout are organized so that the selected cells along the critical path can be identified through the naming criteria for Calibre extraction, which is the default setup used for PEX in the CMOS PDK. Then, the via connections (extracted as resistance instance) from access transistor drain to BL for these cells are replaced with ReRAM device models after PEX is complete.

It can be observed that the CNT technology PDK demonstrates a lower transistor on-off ratio compared to silicon [27]. The resulting higher leakage through transistors (especially through high-density bit-cell arrays) can greatly degrade power metrics for volatile memories such as SRAM. However, the non-volatile characteristic of ReRAM memories avoids this impact and similar read/write energy for CNT and CMOS technologies can be observed. A higher reduction in read latency for the design in the CNT PDK can be attributed to both (1) faster BL discharge from un-selected-cell sneak currents in CNT and (2) Sense Amplifier design optimization. While the sneak current does reduce read latency, it also degrades read margin. The LRS read current is measured and compared with HRS read current to evaluate the impact of subarray level sneak current for both technologies and reported as **read current ratio (RCR)** in Table 2 ($RCR = I_{LRS,subarray}/I_{HRS,subarray}$; $I_{LRS,subarray} = I_{LRS,bit-cell} + (N_{WL} - 1) \times I_{sneak,bit-cell}$; $I_{HRS,subarray} = I_{HRS,bit-cell} + (N_{WL} - 1) \times I_{sneak,bit-cell}$). This metric is a measure of the read margin and a higher value means a better read. It can be seen that the high transistor on-off ratio in the CMOS PDK presents much better RCR. Furthermore, the resulting reduction in read margin from sneak current for the CNT PDK will worsen as subarray dimensions (WL count, N_{WL} , in particular) increase. This can place a fundamental limit on the maximum “readable” subarray size. A comparison of the implemented subarrays is also shown in Table 2 (right).

Table 3. Compilation Time for Subarray Generation and Evaluation

Operation	Pure Generation	Performance Evaluation	Performance Evaluation w/o layout reduction
Layout Generation	~30 s	~30 s	~2 m
Parasitic Extraction	-	~2 m	~5 m
Testbench Processing	-	~30 s	~30 s
Netlist Export	-	~2 m	~10 m
Transient Simulation	-	~3 h	>24 h
Export LEF/GDS/CSV	~30 s	~5 s	~5 s

Fig. 16. Summary and visual view of subarray structures, demonstrated with $64\text{WL} \times 64\text{BL}$ subarrays.

5.2 Subarray Compilation Results

The typical compilation time for generating/exporting layouts of subarrays is under 1 min, where the majority is consumed to load the corresponding tools used. For performance extraction, similar overhead is observed for creating the reduced four-column/four-row layouts. Parasitic extraction overhead is around 2 min for a $64\text{WL} \times 64\text{BL}$ subarray but increases with subarray dimensions, slight increase is observed for MTF subarrays compared to STF subarrays due to increased layer occupied. Transient simulation time also scales with subarray size. For a $64\text{WL} \times 64\text{BL}$ subarray, the simulation time is approximately 3 h for STF subarrays and 5 h for MTF subarrays (tested on a platform with i7gen9 core and 8 GB RAM). However, these simulations only need to be run once for each subarray structure/dimension and are stored in **comma-separated values (CSV)** format within the tool. Table 3 shows a summary of compilation time.

Figure 16 shows a summary of the subarray architectures supported by the compiler and associated layout top and side views. Metal layers are left out in side view plots to present a clearer view of transistor usage in each tier. The footprint overhead introduced by peripherals can be observed from the layout top views presented.

Footprint Analysis. Figure 17(a) shows the 2D footprint area for different subarray dimensions. As expected, both MTF designs show lower footprint than the STF design. However, the benefits are lower for larger subarrays. This is because the relative area of peripheral circuits compared to the bit-cell array is lower for larger subarrays. MTF-BL designs reduce footprint of subarrays close to the minimum required area of bit cells, the only overhead is due to the WL drivers in the same tier of bit cells. In principle, MTF-ALL should show the minimum footprint as all peripherals are folded on top of a single bit-cell array. This is observed for larger subarrays like $128\text{WL} \times 128\text{BL}$.

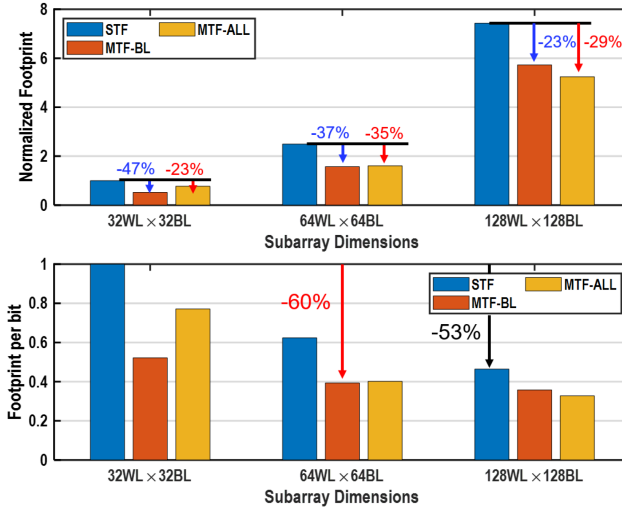


Fig. 17. Normalized footprint (top) and per-bit footprint (bottom) for different subarray sizes.

However, for a $32\text{WL} \times 32\text{BL}$ subarrays, the MTF-ALL shows a higher footprint than MTF-BL. This is because the BL drivers extend beyond the bit cells when the both BL and WL peripherals are stacked as outlined in the top view of Figure 16. If the bit-cell characteristics allow smaller BL drivers for writing, then the footprint of MTF-ALL can be reduced further.

Cell-density Analysis. Figure 17(b) shows the area per bit (bit-cell density) for different subarray architectures and dimensions. The cell density can be increased by $2\times$ by increasing STF subarray dimensions from $32\text{WL} \times 32\text{BL}$ to $128\text{WL} \times 128\text{BL}$. However, MTF-ReRAMs with $64\text{WL} \times 64\text{BL}$ can achieve similar bit-cell density as $128\text{WL} \times 128\text{BL}$ STF ReRAM. The cell density is a key metric for area-efficient memory design. In traditional 2D structures, the reduced cell-density limits the usage of small subarrays. However, smaller subarrays have better performance (lower read latency). Hence, our ReRAM compiler can generate high-density and high-performance subarrays by using MTF-BL or MTF-ALL architecture.

Read Latency and Energy Analysis. Figure 18 compares the read performance and read/write energy per bit of different types of subarrays. The data is normalized to the energy and latency of STF subarrays at the same dimensions. We observe that the MTF designs show lower read latency and read energy compared to the STF subarray. The reductions can be attributed to lower parasitics and reduced IR drop. Moreover, a higher reduction is observed for larger subarrays. Further, we observe MTF-ALL structures produce lower read energy and latency due to central placement of peripherals on the top tier, which minimizes worst-case access paths. The difference is more pronounced for the $128\text{WL} \times 128\text{BL}$ subarray.

Write Energy Analysis. As the write energy is dominated by high current requirement for programming the ReRAM, there is negligible difference between STF and MTF subarray structures. We observe a marginal increase in write energy for MTF options for the $32\text{WL} \times 32\text{BL}$ subarray. This is due to the higher parasitics introduced to the gate of the driver circuits while stacking. This parasitic is noticeable for the write drivers due to the larger size required to pass the write current. and the footprint is larger than precharge circuits for read. However, this overhead

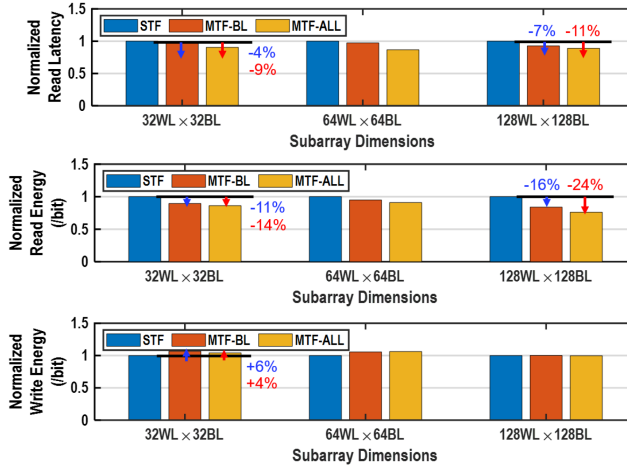


Fig. 18. Read latency and Read/Write energy per-bit comparison for different subarrays (normalized to 2D baseline).

Type	Single Tier	Stacked Peripheral	
	STF	MTF-BL	MTF-ALL
Top view			
Top Side View	Empty!		
Bottom Side View			

Fig. 19. Summary and visual view of full-array structures. Total 32KB and consists of 16 subarrays.

becomes negligible for 64WL × 64BL and 128WL × 128BL designs and hence, all designs show almost the same write energy.

5.3 Array Compilation Results

We generate 32KB array using subarrays of different types and dimensions. The bus width of all designs of memory was fixed at 32 bits. The frequency of the top module is synthesized with the subarray’s maximum frequency (Figure 18). Figure 19 shows example layouts 32KB arrays generated using 128WL × 128BL subarrays of STF, MTF-BL, and MTF-ALL ReRAM. Compilation time of full arrays depends on the depth of the hierarchy (determined by target capacity and subarray dimensions). Arrays of same capacity that are formed with larger subarrays can be generated in a shorter period of time.

Area Analysis. Figure 20 shows the normalized footprint of the 32KB array for different subarrays. As expected, using subarrays of higher dimension results in lower area for the 32KB array. For a given memory size, use of larger subarrays requires fewer levels in the H-tree and, hence, needs less area for logic and routing. Moreover, the arrays designed with MTF subarrays show lower area than the ones designed with STF subarrays.

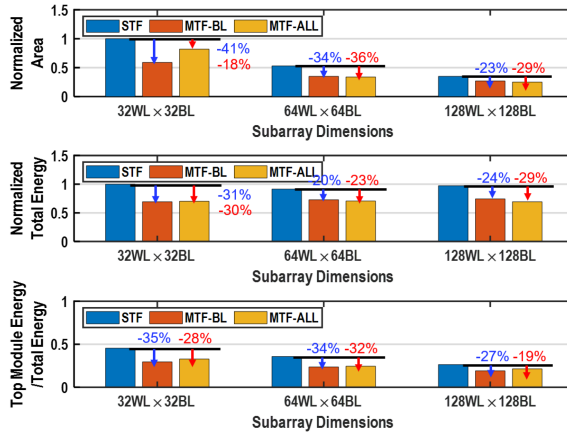


Fig. 20. Array (32KB) analysis showing area (top), energy (middle), and contribution of top module energy to total array access energy (bottom).

Energy Analysis. Figure 20 shows the read energy of the 32KB array, as well as the fraction of the energy consumed by the subarrays. The total array energy is normalized to the array designed using $32\text{WL} \times 32\text{BL}$ subarrays of STF-ReRAM. The read energy of subarrays increases with higher dimension (longer BL/WL implies more parasitics). Moreover, larger subarrays lead to fewer levels in the H-tree and hence, less energy contribution from the top module connections. Therefore, larger subarrays contribute a higher fraction of the total energy. However, as the minimum energy design of the subarray depends on the optimal trade-off between subarray and top module energy. Moreover, MTF-BL and MTF-ALL show a lower energy compared to STF design. This is similar to the trends observed at the subarray level analysis in Figure 18. It is interesting to note the percentage reduction of energy with MTF subarrays are more pronounced at the array level. This is because a smaller footprint of the subarrays designed using MTF architectures reduces routing at the top module (inter-subarray access logic), which further reduces total array power. Consequently, top modules contribute less to the total array energy (Figure 20). This improvement is more pronounced with smaller-dimension subarrays, where more subarrays require higher routing complexity, and the top module logic energy (per access) is close to subarray access energy. It can also be observed that while MTF-ALL subarrays present the least footprint (area), array control power is reduced the most with MTF-BL subarrays at $128\text{WL} \times 128\text{BL}$ ($0.76 \times 0.73 < 0.71 \times 0.81$). This is due to the designed single-side-access for MTF-BL subarrays even in a Multi-tier transistor structure. The advantage of avoiding higher routing overhead for multi-side access with MTF-ALL structures as subarray dimensions increase is successfully captured by the memory compiler for access energy optimization.

6 CONCLUSION

This article demonstrates a ReRAM memory compiler for M3D technologies with multiple tiers of transistors and BEOL ReRAM. The compiler supports automated generation of scalable subarrays with single and multiple tiers of transistors for access and peripheral circuits, as well as large memory modules with multiple interconnected subarrays. Our analysis shows that using multiple tiers of transistors within a ReRAM subarray can lead to reduced footprint and improved read performance/energy, enabling generation of high-density and high-performance ReRAM macros. Different MTF structures can benefit different metrics depending on the target application. Our

compiler generates layout (LEF/GDS), schematic, and symbol views in a fully automated fashion that can be directly used directly for chip-scale physical design. CSV and LIB files are generated for subarray and array performance, respectively. As M3D process and ReRAM technologies continue to mature, future work will couple our compiler with M3D logic design tools, to design, evaluate, and optimize new system architecture and generate full-chip physical designs.

REFERENCES

- [1] Aya G. Amer, Rebecca Ho, Gage Hills, Anantha P. Chandrakasan, and Max M. Shulaker. 2019. 29.8 SHARC: Self-healing analog with RRAM and CNFETs. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC'19)*. 470–472.
- [2] Perrine Batude, Thomas Ernst, Julien Arcamone, Gregory Arndt, Perceval Coudrain, and Pierre-Emmanuel Gaillardon. 2012. 3D sequential integration: A key enabling technology for heterogeneous co-integration of new function with CMOS. *IEEE J. Emerg. Select. Top. Circ. Syst.* 2, 4 (2012), 714–722.
- [3] Mindy D. Bishop, Gage Hills, Tathagata Srimani, Christian Lau, Denis Murphy, Samuel Fuller, Jefford Humes, Anthony Ratkovich, Mark Nelson, and Max M. Shulaker. 2020. Fabrication of carbon nanotube field-effect transistors in commercial silicon manufacturing facilities. *Nature Electron.* 3, 8 (Aug 2020), 492–501.
- [4] Meng-Fan Chang, Chien-Chen Lin, Albert Lee, Chia-Chen Kuo, Geng-Hau Yang, Hsiang-Jen Tsai, Tien-Fu Chen, Shyh-Shyuan Sheu, Pei-Ling Tseng, Heng-Yuan Lee, and Tzu-Kun Ku. 2015. 17.5 A 3T1R nonvolatile TCAM using MLC ReRAM with Sub-1ns search time. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC'15)*.
- [5] Meng-Fan Chang, Che-Wei Wu, Chia-Cheng Kuo, Shin-Jang Shen, Ku-Feng Lin, Shu-Meng Yang, Ya-Chin King, Chong-Jung Lin, and Yu-Der Chih. 2012. A 0.5V 4Mb logic-process compatible embedded resistive RAM (ReRAM) in 65 nm CMOS using low-voltage current-mode sensing scheme with 45ns random read time. In *Proceedings of the IEEE International Solid-State Circuits Conference*. 434–436.
- [6] Meng-Fan Chang, Jui-Jen Wu, Tun-Fei Chien, Yen-Chen Liu, Ting-Chin Yang, Wen-Chao Shen, Ya-Chin King, Chong Jung Lin, Ku-Feng Lin, Yu-Der Chih, and Jonathan Chang. 2015. Low VDDmin swing-sample-and-couple sense amplifier and energy-efficient self-boost-write-termination scheme for embedded ReRAM macros against resistance and switch-time variations. *IEEE J. Solid-State Circ.* 50, 11 (2015), 2786–2795.
- [7] Wei-Hao Chen, Kai-Xiang Li, Wei-Yu Lin, Kuo-Hsiang Hsu, Pin-Yi Li, Cheng-Han Yang, Cheng-Xin Xue, En-Yu Yang, Yen-Kai Chen, Yun-Sheng Chang, Tzu-Hsiang Hsu, Ya-Chin King, Chong-Jung Lin, Ren-Shuo Liu, Chih-Cheng Hsieh, Kea-Tiong Tang, and Meng-Fan Chang. 2018. A 65 nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC'18)*. 494–496.
- [8] Chiyui Ahn, Zizhen Jiang, Chi-Shuen Lee, Hong-Yu Chen, J. Liang, L. S. Liyanage, and H. P. Wong. 2014. A 1TnR array architecture using a one-dimensional selection device. In *Proceedings of the Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers*. 1–2.
- [9] Sujin Choi, Wookyoung Sun, and Hyungsoon Shin. 2018. Analysis of read margin and write power consumption of a 3-D vertical RRAM (VRRAM) crossbar array. *IEEE J. Electron. Devices Soc.* 6 (2018), 1192–1196.
- [10] Jason Cong, Guojie Luo, Jie Wei, and Yan Zhang. 2007. Thermal-aware 3D IC placement via transformation. In *Proceedings of the Asia and South Pacific Design Automation Conference*. 780–785.
- [11] C. Fenouillet-Beranger, B.Mathieu, B. Previtali, M-P. Samson, N.Rambal, V. Benevent, S. Kerdiles, J-P. Barnes, D. Barge, P. Besson, R. Kachtouli, M. Casse, X. Garros, A. Laurent, F. Nemouchi, K. Huet, I. Toque-Tresonne, D.Lafond, H. Dansas, F. Aussenac, G. Druais, P. Perreau, E. Richard, S. Chhun, E. Petitprez, N. Guillot, F. Deprat, L. Pasini, L. Brunet, V. Lu, C. Reita, P. Batude, and M. Vinet. 2014. New insights on bottom layer thermal stability and laser annealing promises for high performance 3D VLSI. In *Proceedings of the IEEE International Electron Devices Meeting*. 27.5.1–27.5.4.
- [12] Gage Hills, Marie Garcia Bardon, Gerben Doornbos, Dmitry Yakimets, Pieter Schuddinck, Rogier Baert, Doyoung Jang, Luca Mattii, Syed Muhammed Yasser Sherazi, Dimitrios Rodopoulos, Romain Ritzenthaler, Chi-Shuen Lee, Aaron Voon-Yew Thean, Iuliana Radu, Alessio Spessot, Peter Debacker, Francky Catthoor, Praveen Raghavan, Max M. Shulaker, H.-S. Philip Wong, and Subhasish Mitra. 2018. Understanding energy efficiency benefits of carbon nanotube field-effect transistors for digital VLSI. *IEEE Trans. Nanotechnol.* 17, 6 (2018), 1259–1269.
- [13] R. Ishihara, M. R. T. Mofrad, J. Derakhshandeh, N. Golshani, and C. I. M. Beenakker. 2012. Monolithic 3D-ICs with single grain Si thin film transistors. In *Proceedings of the IEEE 11th International Conference on Solid-State and Integrated Circuit Technology*. 1–4.

- [14] Sureshwar Kannan, Rahul Agarwal, Arnaud Bousquet, Geetha Aluri, and Hui-Shan Chang. 2015. Device performance analysis on 20 nm technology thin wafers in a 3D package. In *Proceedings of the IEEE International Reliability Physics Symposium (IRPS'15)*.
- [15] B. W. Ku, Kyungwook Chang, and Sung Kyu Lim. 2020. Compact-2D: A physical design methodology to build two-tier gate-level 3D ICs. *IEEE Trans. Comput.-Aided Design* 39, 6 (2020), 1151–1164.
- [16] Albert Lee, Chieh-Pu Lo, Chien-Chen Lin, Wei-Hao Chen, Kuo-Hsiang Hsu, Zhibo Wang, Fang Su, Zhe Yuan, Qi Wei, Ya-Chin King, Chrong-Jung Lin, Hochul Lee, Pedram Khalili Amiri, Kang-Lung Wang, Yu Wang, Huazhong Yang, Yongpan Liu, and Meng-Fan Chang. 2017. A ReRAM-based nonvolatile flip-flop with self-write-termination scheme for frequent-OFF fast-wake-up nonvolatile processors. *IEEE J. Solid-State Circ.* 52, 8 (2017), 2194–2207.
- [17] Yongpan Liu, Zhibo Wang, Albert Lee, Fang Su, Chieh Pu Lo, Zhe Yuan, Chien Chen Lin, Qi Wei, Yu Wang, Ya Chin King, Chrong Jung Lin, Pedram Khalili, Kang Lung Wang, Meng Fan Chang, and Huazhong Yang. 2016. 4.7 A 65 nm ReRAM-enabled nonvolatile processor with 6× reduction in restore time and 4× higher clock frequency using adaptive data retention and self-write-termination nonvolatile logic. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC'16)*. 84–86.
- [18] Yi-Chen Lu, Sai Surya Kiran Pentapati, Lingjun Zhu, Kambiz Samadi, and Sung Kyu Lim. 2020. TP-GNN: A graph neural network framework for tier partitioning in monolithic 3D ICs. In *Proceedings of the ACM Design Automation Conference (DAC'20)*.
- [19] H. Henry Nho, Mark Horowitz, and S. Simon Wong. 2008. A high-speed, low-power 3D-SRAM architecture. In *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC'08)*. 201–204.
- [20] Heechun Park, Kyungwook Chang, Bon Woong Ku, Jinwoo Kim, Edward Lee, Daehyun Kim, Arjun Chaudhuri, Sammitra Banerjee, Saibal Mukhopadhyay, Krishnendu Chakrabarty, and Sung Kyu Lim. 2019. RTL-to-GDS tool flow and design-for-test solutions for monolithic 3D ICs. In *Proceedings of the ACM Design Automation Conference (DAC'19)*.
- [21] Sandeep Kumar Samal, Deepak Nayak, Motoi Ichihashi, Srinivasa Banna, and Sung Kyu Lim. 2016. How to cope with slow transistors in the top tier of monolithic 3D ICs: Design studies and CAD solutions. In *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED'16)*. 320–325.
- [22] Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramonian, John Paul Strachan, Miao Hu, R. Stanley Williams, and Vivek Srikumar. 2016. ISAAC: A convolutional neural network accelerator with in situ analog arithmetic in crossbars. In *Proceedings of the International Symposium on Computer Architecture (ISCA'16)*. 14–26.
- [23] M. M. Shulaker et al. 2014. Monolithic three-dimensional integration of carbon nanotube FETs with silicon CMOS. In *Proceedings of the Symposium on VLSI Technology (VLSI-Technology'14)*. 1–2.
- [24] Max M. Shulaker, Gage Hills, Rebecca S. Park, Roger T. Howe, Krishna Saraswat, H.-S. Philip Wong, and Subhasish Mitra. 2017. Three-dimensional integration of nanotechnologies for computing and data storage on a single chip. *Nature* 547 (2017), 74–78.
- [25] Max M. Shulaker, Tony F. Wu, Asish Pal, Liang Zhao, Yoshio Nishi, Krishna Saraswat, H.-S. Philip Wong, and Subhasish Mitra. 2014. Monolithic 3D integration of logic and memory: Carbon nanotube FETs, resistive RAM, and silicon FETs. In *Proceedings of the IEEE International Electron Devices Meeting*. 27.4.1–27.4.4.
- [26] Max M. Shulaker, Tony F. Wu, Mohamed M. Sabry, Hai Wei, H.-S. Philip Wong, and Subhasish Mitra. 2015. Monolithic 3D integration: A path from concept to reality. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE'15)*. 1197–1202.
- [27] T. Srimani, G. Hills, M. Bishop, C. Lau, P. Kanhaiya, R. Ho, A. Amer, M. Chao, A. Yu, A. Wright, A. Ratkovich, D. Aguilar, A. Bramer, C. Cecman, A. Chov, G. Clark, G. Michaelson, M. Johnson, K. Kelley, P. Manos, K. Mi, U. Suriono, S. Vuntangboon, H. Xue, J. Humes, S. Soares, B. Jones, S. Burack, Arvind, A. Chandrakasan, B. Ferguson, M. Nelson, and M. M. Shulaker. 2020. Heterogeneous integration of BEOL logic and memory in a commercial foundry: Multi-tier complementary carbon nanotube logic and resistive RAM at a 130 nm node. In *Proceedings of the IEEE VLSI Symposium on Technology and Circuits*.
- [28] Tathagata Srimani, Gage Hills, Mindy Deanna Bishop, and Max M. Shulaker. 2019. 30-nm contacted gate pitch back-gate carbon nanotube FETs for sub-3-nm nodes. *IEEE Trans. Nanotechnol.* 18 (2019), 132–138.
- [29] Srivatsa Srinivasa, Akshay Krishna Ramanathan, Xueqing Li, Wei-Hao Chen, Sumeet Kumar Gupta, Meng-Fan Chang, Swaroop Ghosh, Jack Sampson, and Vijaykrishnan Narayanan. 2018. A monolithic-3D SRAM design with enhanced robustness and in-memory computation support. In *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED'18)*.
- [30] M. Vinet, P. Batude, C. Fenouillet-Beranger, F. Clermidy, L. Brunet, O. Rozeau, J. M. Hartmann, O. Billoint, G. Cibrario, B. Previtali, C. Tabone, B. Sklenard, O. Turkyilmaz, F. Ponthenier, N. Rambal, M. P. Samson, F. Deprat, V. Lu, L. Pasini, S. Thuries, H. Sarhan, J.-E. Michallet, and O. Faynot. 2014. Monolithic 3D integration: A powerful alternative to classical 2D scaling. In *Proceedings of the SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S'14)*. 1–3.

- [31] H.-S. Philip Wong, Heng-Yuan Lee, Shimeng Yu, Yu-Sheng Chen, Yi Wu, Pang-Shiu Chen, Byoungil Lee, Frederick T. Chen, and Ming-Jinn Tsai. 2012. Metal–Oxide RRAM. *Proc. IEEE* 100, 6 (2012), 1951–1970.
- [32] Tony F. Wu, Haitong Li, Ping-Chen Huang, Abbas Rahimi, Gage Hills, Bryce Hodson, William Hwang, Jan M. Rabaey, H.-S. Philip Wong, Max M. Shulaker, and Subhasish Mitra. 2018. Hyperdimensional computing exploiting carbon nanotube FETs, resistive RAM, and their monolithic 3D integration. *IEEE J. Solid-State Circ.* 53, 11 (2018), 3183–3196.

Received September 2020; revised February 2021; accepted May 2021